

NIKOFLIX – Project Design & Implementation Report

1. Project Overview

NIKOFLIX is an OTT-style video streaming platform designed to support movies, series, episodic content, user engagement, and monetization. The project emphasizes strong database design, realistic system modeling, and real-world scalability principles used in production-grade streaming platforms.

2. Technology Stack

The project is built using the PERN stack: PostgreSQL – relational database with strong constraints Node.js & Express – backend API layer React – frontend interface PostgreSQL was chosen for its support of data integrity, relationships, and transactional reliability.

3. Identity & User Model

Every interaction in the system is associated with an identity. An identity can either be a registered user or a guest user. Guest identities allow anonymous interaction, while registered users have permanent accounts. Only one identity type is active per request, ensuring consistency and clean data separation.

4. Authentication & Verification

Users can register using an email address or a phone number. Authentication supports password-based login along with OTP verification. OTPs are used for account verification, alternate login flows, and account recovery. This design provides flexibility while maintaining security.

5. Content Architecture

All content is modeled using a unified content structure. Content is categorized as either a movie or a series. Related content such as sequels or seasons are grouped using a parent-child relationship. The first released item acts as the anchor for grouping, allowing logical association without unnecessary abstraction.

6. Playable Media Units

Movies and episodes are stored as separate playable entities. Movies represent standalone content, while episodes belong to series. Each playable unit stores duration, streaming URL, and view-related metadata, enabling accurate playback and tracking.

7. Watch History & Views

Watch history tracks playback activity for both guest and registered users. Views are calculated as unique viewers based on identity, ensuring replays do not inflate view counts. This approach reflects industry-standard analytics practices.

8. User Interactions

Users can interact with content through likes, comments, and wishlists. Guests are allowed to like content, while commenting requires a registered account. This balances engagement with accountability. Wishlists are restricted to registered users for persistent personalization.

9. Payments & Monetization

The system supports both subscription-based access and one-time content unlocks. Payments act as the source of truth for access control. Each transaction is recorded with gateway references and

status tracking, ensuring reliable monetization handling.

10. Trending System

Trending content is managed using a rank-based system. Content is categorized by type and assigned a rank that represents popularity. This structure allows efficient display of trending movies and series without expensive real-time computation.

11. Database Integrity

The database schema enforces strict constraints such as identity exclusivity, valid content relationships, and unique interactions. These constraints prevent inconsistent data and reflect production-grade database design standards.

12. Conclusion

NIKOFLIX has been designed with a strong focus on realism, scalability, and correctness. The system architecture and database schema align closely with real-world streaming platforms while remaining flexible for future enhancements. The project demonstrates a clear understanding of backend design principles and modern OTT system requirements.